Perturb-and-Revise: Flexible 3D Editing with Generative Trajectories

Supplementary Material

A. Score Distillation as Particle-Based Variational Inference

Our parameter perturbation and identity gradients build upon the mathematical intuition of the variational score distillation (VSD) approach [15], an extension of score distillation sampling (SDS) [10]. In this context, the parameters of NeRF during distillation are treated as particles.

VSD minimizes the KL divergence between a variational distribution $q^\gamma(x|c)$, which is implicitly modeled by γ, and the target distribution $p_\phi(x|c)$, which is implicitly modeled by the diffusion model ϕ. Incorporating timesteps and camera poses, the objective is formulated as follows:

$$\gamma^* := \arg\min_\gamma \mathbb{E}_{t,\psi}\Big[\frac{\sigma_t}{\alpha_t} w(t) D_{\mathrm{KL}}(q_t^\gamma(x_t|c,t)\|p_\phi(x_t|c,t))\Big] \tag{1}$$

where $\frac{\sigma_t}{\alpha_t}$ and $w(t)$ are diffusion-related weighting factors, and $q_t^\gamma(x_t|c,t)$ and $p_\phi(x_t|c,t)$ represent the distributions of noisy images to be modeled by diffusion models.

To minimize this objective, VSD employs particle-based variational inference based on Wasserstein gradient flow, as detailed in [1, 2, 8, 14]. Specifically, the Wasserstein gradient flow satisfies:

$$\frac{\partial \gamma_\tau}{\partial \tau} = \nabla \cdot (\gamma_\tau \nabla (\frac{\partial E}{\partial \gamma_\tau}(\gamma_\tau))) \tag{2}$$

In our case, the energy functional E is defined as follows:

$$E(\gamma) := \mathbb{E}_{t,\psi}\Big[\frac{\sigma_t}{\alpha_t} w(t) D_{\mathrm{KL}}(q_t^\gamma(x_t|c,t)\|p_\phi(x_t|c,t))\Big] \tag{3}$$

In the particle-based variational inference, particles represent samples from the variational distribution. A set of M particles $\{\theta^{(i)}\}_{i=1}^M \sim \gamma$ is iteratively updated following the velocity of particles [1]: $\frac{d\theta_\tau}{d\tau} = \nabla(\frac{\partial E}{\partial \gamma_\tau}(\gamma_\tau))$. With the energy function in Eq. 3, the particles follow the ordinary differential equation (ODE):

$$\frac{d\theta_\tau}{d\tau} = -\,\mathbb{E}_{t,\epsilon,\psi}\Big[w(t)\big(-\sigma_t \nabla_{x_t} \log p_\phi(x_t|c,t) - (-\sigma_t \nabla_{x_t} \log q_t^{\gamma_\tau}(x_t|c,t))\frac{\partial g(\theta_\tau,\psi)}{\partial \theta_\tau}\big)\Big] \tag{4}$$

where τ denotes the ODE time, constrained to $\tau \geq 0$, and γ_τ progressively evolves toward the optimal distribution γ^* as $\tau \to \infty$. In this VSD framework, the gradient of the SDS loss is a specific instance of the equation [15], where a single particle represents the entire distribution.

B. Resulting Distribution from Parameter Interpolation (Sec. 4.1)

Here, we show that interpolating parameters with $\eta \in [0, 1]$ results in a versatile sampling distribution that interpolates between a point mass at θ_{src} and the initial distribution. Given a source parameter θ_{src} and an initial distribution $\mathcal{P}(\Theta_0)$ with bounded variance σ^2, we define the parameter perturbation as:

$$\theta_{\text{perturbed}} = (1 - \eta)\theta_{\text{src}} + \eta\theta_0, \quad \theta_0 \sim \mathcal{P}(\Theta_0), \quad \eta \in [0, 1] \tag{5}$$

Using the change of variables formula with transformation $T(\theta_0) = (1 - \eta)\theta_{\text{src}} + \eta\theta_0$ and its inverse $T^{-1}(\theta_{\text{perturbed}}) = (\theta_{\text{perturbed}} - (1 - \eta)\theta_{\text{src}})/\eta$:

$$p(\theta_{\text{perturbed}}) = \mathcal{P}(\Theta_0)(T^{-1}(\theta_{\text{perturbed}})) \cdot |\det(J_{T^{-1}})| \tag{6}$$

Since the Jacobian matrix is $J_{T^{-1}} = \frac{1}{\eta} I_d$, where I_d is the d-dimensional identity matrix, we have:

$$p(\theta_{\text{perturbed}}) = \frac{1}{\eta^d} \mathcal{P}(\Theta_0) \left(\frac{\theta_{\text{perturbed}} - (1 - \eta)\theta_{\text{src}}}{\eta} \right) \tag{7}$$

Here, η controls the degree of interpolation through both a scale factor $\frac{1}{\eta^d}$ and the argument $(\theta_{\text{perturbed}} - (1 - \eta)\theta_{\text{src}})/\eta$ of $\mathcal{P}(\Theta_0)$.

For $\eta \to 1$, both terms approach simple limits:

$$\lim_{\eta \to 1} p(\theta_{\text{perturbed}}) = \lim_{\eta \to 1} \frac{1}{\eta^d} \mathcal{P}(\Theta_0) \left(\frac{\theta_{\text{perturbed}} - (1 - \eta)\theta_{\text{src}}}{\eta} \right) \tag{8}$$

$$= \mathcal{P}(\Theta_0)(\theta_{\text{perturbed}}) \tag{9}$$

For $\eta \to 0$, we consider the distribution of $\theta_{\text{perturbed}}$. By Chebyshev's inequality, for any $\varepsilon > 0$:

$$P(|\theta_{\text{perturbed}} - \mathbb{E}[\theta_{\text{perturbed}}]| \geq \varepsilon) \leq \frac{\eta^2 \sigma^2}{\varepsilon^2} \to 0 \quad \text{as} \quad \eta \to 0 \tag{10}$$

Moreover, since $\mathbb{E}[\theta_{\text{perturbed}}] \to \theta_{\text{src}}$ as $\eta \to 0$:

$$P(|\theta_{\text{perturbed}} - \theta_{\text{src}}| \geq \varepsilon) \to 0 \quad \text{as} \quad \eta \to 0 \tag{11}$$

This proves convergence in probability to θ_{src}. The $\frac{1}{\eta^d}$ factor ensures that the total probability remains 1, while the concentration around θ_{src} becomes arbitrarily tight as $\eta \to 0$, characterizing convergence to:

$$\lim_{\eta \to 0} p(\theta_{\text{perturbed}}) = \delta(\theta_{\text{perturbed}} - \theta_{\text{src}}) \tag{12}$$

Thus, we have shown that the interpolation of parameters results in an interpolation between two extremes: a point mass at θ_{src} and the initial distribution, and the parameter η controls the degree of interpolation, i.e., the versatility.

Algorithm 1: Parameter Perturbation

Function `ParameterPerturbation(`η`)`**:**
- $\theta_{\text{new}} \leftarrow$ Initialize new geometry instance
- **for** $(\theta_c, \theta_n, \theta_i)$ ***in*** *zip($\theta_{current}$, θ_{new}, θ_{init})* **do**
  - $\theta_c \leftarrow (1-\eta)\theta_i + \eta\theta_n$ `// Parameter interpolation`
- **end**
- Free memory and clear cache

Algorithm 2: Parameter Perturbation with Adaptive η Selection

Input: Empty loss history list $\mathcal{L}$, minimum loss decrease Δ_{min}, maximum parameter perturbation η_{max}
Input: Initial NeRF parameters θ_{init}
Function `TrainingStep`**:**
- **if** $|\mathcal{L}| = 50$ **then**
  - $\Delta\mathcal{L} \leftarrow$ `LossDecrease(`$\mathcal{L}$`)`
  - $\eta \leftarrow$ `DetermineEta(`$\Delta\mathcal{L}$, Δ_{min}, η_{max}`)`
  - `ParameterPerturbation(`η`)`
- **end**
- Proceed with training step
- $\mathcal{L} \leftarrow \mathcal{L} \oplus \{\text{Current step training loss}\}$

Function `LossDecrease(`$\mathcal{L}$`)`**:**
- $\mathcal{L}_{\text{final}} \leftarrow \frac{1}{10}\sum_{i=|\mathcal{L}|-10}^{|\mathcal{L}|} \mathcal{L}_i$ `// Average of last 10 losses`
- $\mathcal{L}_{\text{init}} \leftarrow \frac{1}{10}\sum_{i=1}^{10} \mathcal{L}_i$ `// Average of first 10 losses`
- **return** $\mathcal{L}_{final} - \mathcal{L}_{init}$

Function `DetermineEta(`$\Delta\mathcal{L}$, Δ_{min}, η_{max}`)`**:**
- **return** $\max(0, \eta_{max}(1 - 2^{-(\Delta\mathcal{L}+\Delta_{min})/\Delta_{min}}))$

C. Algorithms for Parameter Perturbation and Adaptive η Selection

We present the complete algorithms for parameter perturbation in Alg. 1 and the adaptive η selection method in Alg. 2. The underlying intuition for the adaptive η selection algorithm is that there exists a minimum loss decrease Δ_{min} required for parameter perturbation and a maximum parameter perturbation η_{max} that can be applied without resulting in complete regeneration of the object. Here, we have two parameters to control, Δ_{min} and η_{max}. Δ_{min} is set to 1000 based on observations that it achieves near-optimal CLIP directional similarity and CLIP directional consistency, as shown in Table 1. η_{max} is set to 0.6 based on the finding that the percentage of successful experiments drops significantly when η exceeds 0.6, as shown in the main paper.

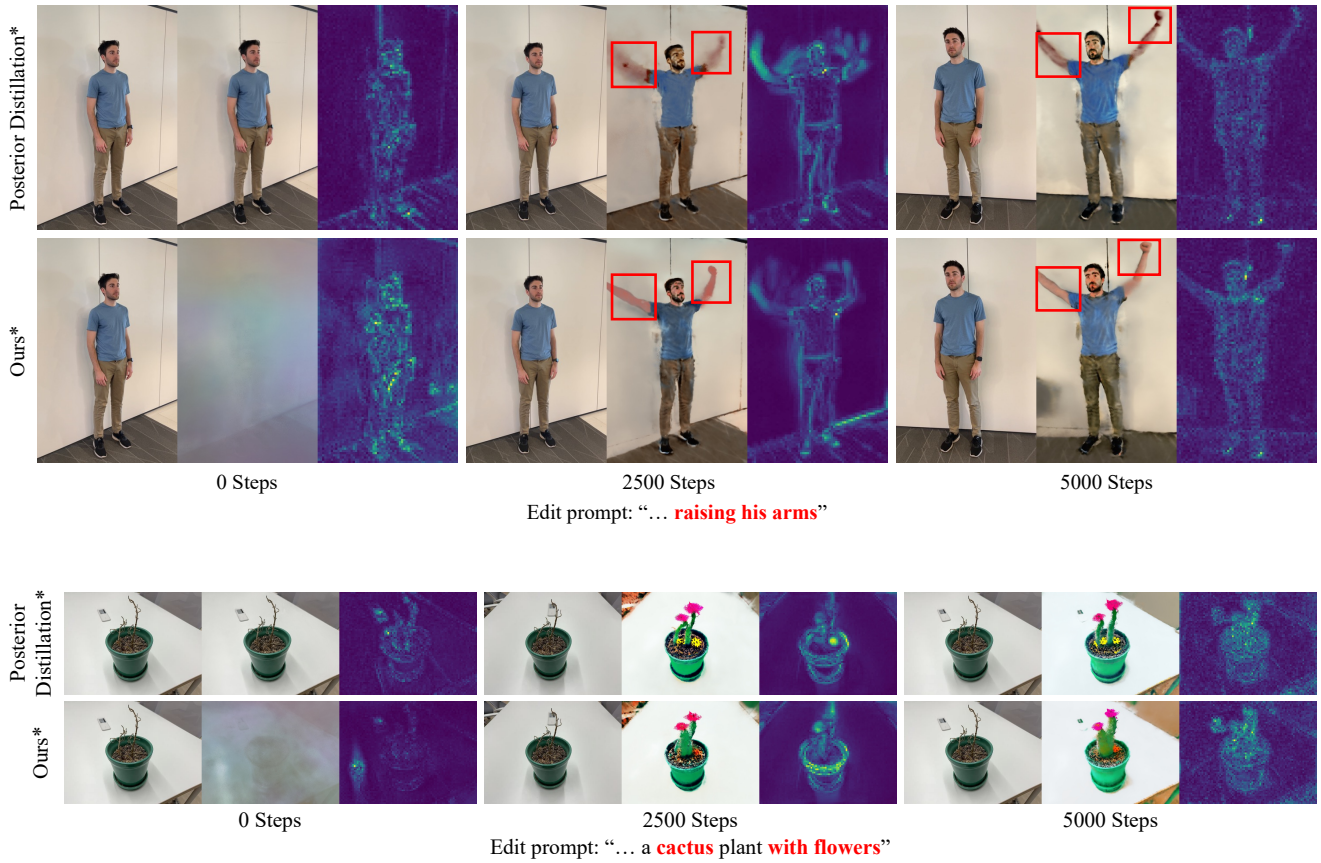


Figure 1. Original scene, edited scene, and image-level gradients are shown at 0, 2500, and 5000 optimization steps. We can see that the density forms earlier and changes drastically even when the perturbation is large and barely has any structure.

D. Additional Analyses

Intermediate results from real scene editing. In Fig. 1, we show intermediate results from a real scene editing experiment. In this experiment, we aim to make the person raise the arms. Despite the initialization having little 3D structure, it converges faster with the same number of optimization steps. We can see that the density near the raised arms quickly converges with parameter perturbation, while the original PDS [6] generates blurry results. This demonstrates the effectiveness of our parameter perturbation approach in various editing scenarios.

Additional ablation study. We present additional ablation study results on Δ_{min} in Table 1. Additionally, we examine the effects of different values for λ_{L1} and λ_{p} in Table 2. Our results demonstrate that our method is relatively robust to these parameters, with our chosen values achieving a near-optimal balance across metrics. In addition, in Fig. 6, we display additional visualizations for the selection of η. A-LPIPS [4] is a metric for view consistency between adjacent frames, and CLIP directional consistency [3] is a metric that computes how much the editing directions differ across frames. Considering that we showed in the main paper that using fixed values of $\eta \geq 0.6$ had a higher likelihood of causing errors, our method outperforms approaches using fixed values of $\eta < 0.6$ in both metrics while maintaining lower error rates.

Additional comparisons. In Figs. 4 and 5, we showcase additional comparisons with the baseline methods.

Comparisons in 360° views. We present qualitative comparisons with 360° views on our project page.

Method	CLIP-Dir-Sim$_{\text{averaged}}\uparrow$	CLIP-Dir-Con$_{\text{averaged}}\uparrow$	LPIPS$_{\text{averaged}}\downarrow$
$\Delta_{\text{min}} = 500$	0.061	0.757	0.112
$\Delta_{\text{min}} = 1000$	0.062	0.757	0.115
$\Delta_{\text{min}} = 2000$	0.060	0.754	0.111

Table 1. Experiment controlling Δ_{min}.

Method	CLIP-Dir-Sim$_{\text{averaged}}\uparrow$	CLIP-Dir-Con$_{\text{averaged}}\uparrow$	LPIPS$_{\text{averaged}}\downarrow$
$\lambda_{\text{L1}} = 10000, \lambda_{\text{p}} = 100$	0.057	0.777	0.115
$\lambda_{\text{L1}} = 30000, \lambda_{\text{p}} = 300$	0.057	0.764	0.105
$\lambda_{\text{L1}} = 50000, \lambda_{\text{p}} = 500$	0.051	0.752	0.091

Table 2. Experiment controlling λ_{L1} and λ_{p}.

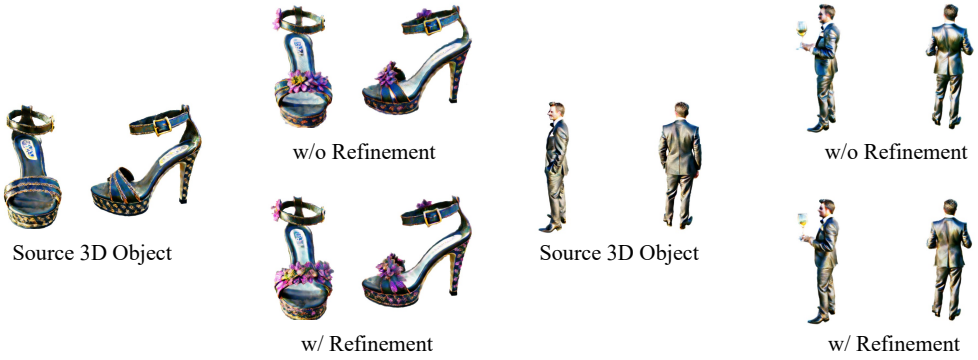


Figure 2. Effects of IPG refinement steps. IPG refinement steps restore changed attributes that were not explicitly mentioned in the edit prompt during the editing process, for example, the support part of the strap and the subtle details in the color and texture of the tuxedo.

Effect of IPG. In Fig. 2, we demonstrate refinement outcomes through IPG and the generative ODE. A notable IPG attribute is its preservation of areas in the 3D object not explicitly specified for modification within the editing prompt.

Extension to 3DGS. Adapting our method to 3DGS [5] presents unique challenges, particularly in addressing adaptive densification and the ill-defined interpolation of initial and optimized Gaussians. However, our preliminary experiment, based on 3DGS generated by LucidDreamer [7], with mean and variance perturbation shows promising results, yielding improved depth maps and geometry under identical conditions (Fig. 3).

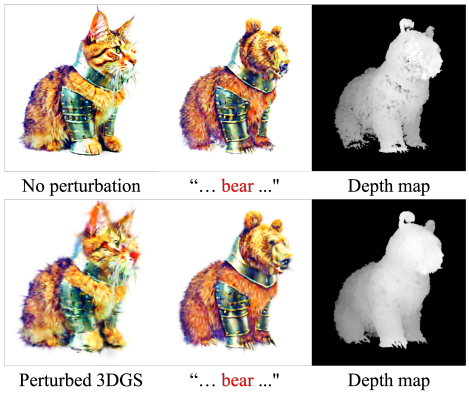


Figure 3. 3DGS experiment. The perturbation approach improves the depth map and overall geometry of the resulting object.

E. Implementation Details

Optimization steps. For all perturbation values, we perform 1.5k editing steps, significantly fewer than the 10k steps required for regeneration [12]. We set a resolution milestone in fashion object editing at which the rendering resolution changes for efficacy to half the number of editing steps. We perform 1k additional refinement steps, making the total runtime similar to 1.5k steps of PDS and thus highly efficient.

Identity-preserving gradients. For the identity-preserving gradients in Sec. 4.3, we adopt a combination of perceptual and L1 losses, finding this more stable and less fragile to noise than using only L1 or L2 loss. Specifically, we choose $\lambda_{\text{L1}} = 300.0$

Figure 4. Additional comparisons of fashion object editing with Score Distillation [10], Posterior Distillation [6], Instruct-NeRF2NeRF [3], and Perturb-and-Revise (ours).

and $\lambda_{\text{p}} = 30000.0$, with an annealed schedule, i.e., we linearly decrease them to 0 until the halfway point of the steps. We present an ablation study on the scales of λ_{p} and λ_{L1} in Table 2.

Timestep annealing. In the original Score Distillation [10], Instruct-NeRF2NeRF [3], and Posterior Distillation [6] papers, a fixed schedule, $\Sigma := \mathcal{U}(0.02, 0.98)$, is utilized. Contrary to this fixed schedule, and considering that our editing purpose does not inherently start from random parameters, we adopt a schedule in which $\Sigma(0) = \mathcal{U}(0.75, 0.75)$, a range to be decreased to $(0.02, 0.4)$ by the time 80% of the total editing steps are reached.

NeRF representation. Technically, the parameter perturbation method can be applied to arbitrary representations whose parameters are initialized from a distribution and optimized. For computational efficiency while maintaining high quality of 3D objects, we choose InstantNGP [9] as our NeRF implementation.

Real scene editing. We show in the main paper that our parameter perturbation approach can be readily extended to real scene editing scenarios [3]. Using Nerfstudio's implementation of Nerfacto [13] as the representation, we integrate Instruct-NeRF2NeRF [3] without modifications. For this experiment, we build upon the distillation method proposed in PDS [6], use Stable Diffusion v1-5 [11] as the backbone, and set $\eta = 0.6$. Besides the perturbation, we use the exact same update rule as PDS. We reduce the timesteps by half and omit the selection and refinement steps for both PDS and our method to manage computational complexity and to show only the effect of the parameter perturbation approach. Note that PDS has an internal preservation term [6], and we use it as is for scene editing. Even with these shortened iteration steps, our parameter perturbation approach enables extensive geometric editing of the scene.

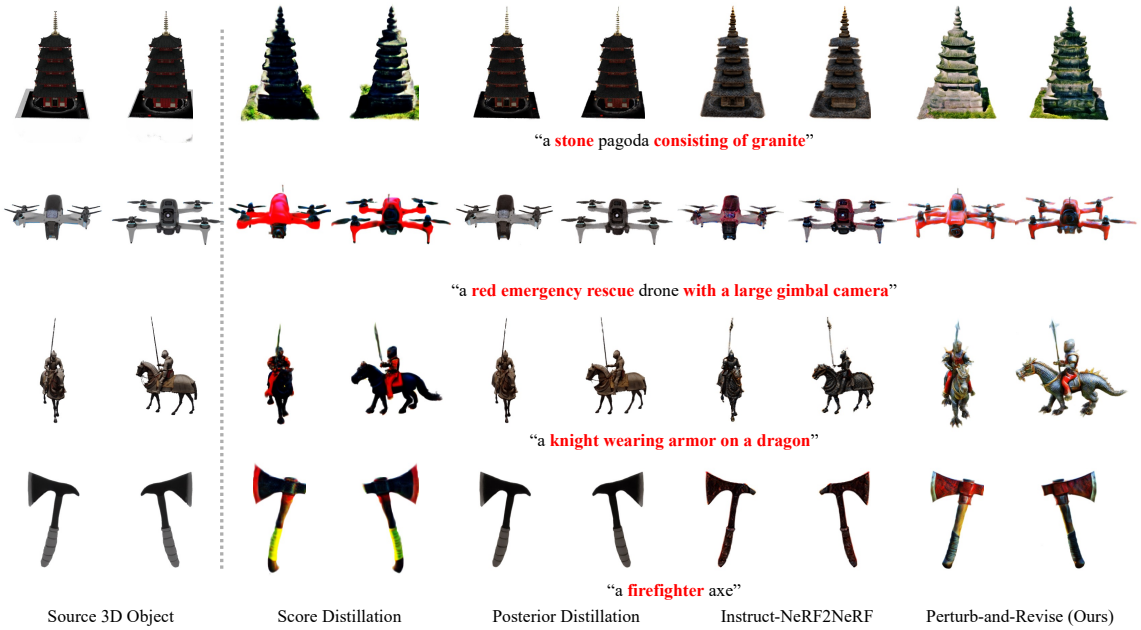
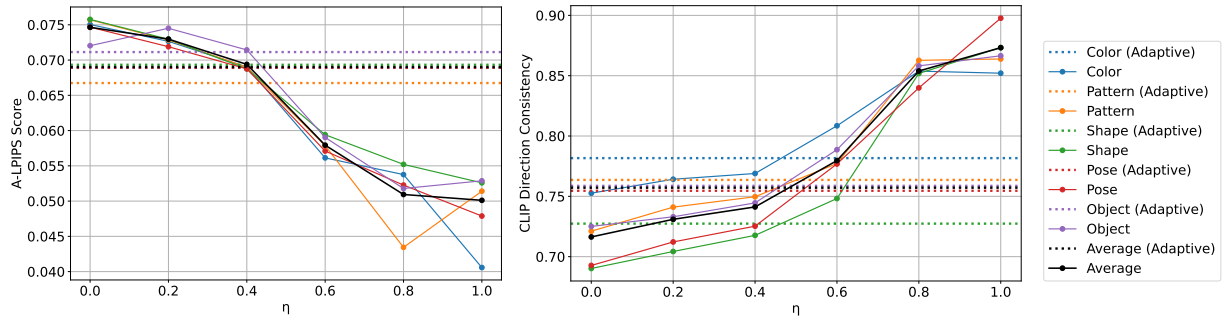
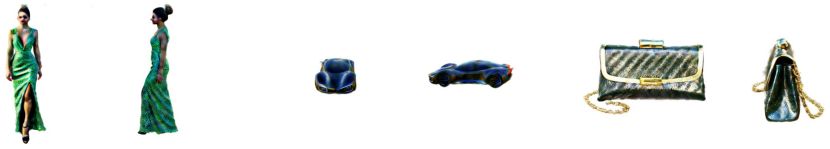


Figure 5. Additional comparisons of general object editing with Score Distillation [10], Posterior Distillation [6], Instruct-NeRF2NeRF [3], and Perturb-and-Revise (ours).



(a) (b)

Figure 6. Additional visualizations for the selection of η. (a) and (b) show the A-LPIPS (lower is better) and CLIP directional consistency (higher is better) for different η values, respectively.



"... a green dress" "... a black alien car" "… a tube of lipstick beside it"

Figure 7. Failure cases.

F. Computational efficiency

Our approach requires approximately 7 minutes with IPG and 4 minutes without, to produce meaningful results. This is faster than the 13 minutes needed for Instruct-NeRF2NeRF. We attribute this to Instruct-NeRF2NeRF requiring updates to the entire dataset, while our method of parameter perturbation and timestep annealing benignly affects the optimization process of an object. Note that completely regenerating an object requires around 26 minutes.

G. Limitations

Although we address limitations of previous work, our method inherits some limitations from pre-trained diffusion models [11, 12], such as color biases and saturation artifacts (Fig. 7). While our method can handle pose and object changes, the compositionality issue of the pre-trained models, which often cannot generate compositions of two or more objects or attributes, is another problem, making our method difficult to accommodate changes to the entire layout.

H. Discussion and Future Work

Perturb-and-Revise (PnR) is a training-free editing method that is fast and effective, opening up new possibilities. The main point of the paper is that parameter perturbation is of prime importance in achieving these results. This approach can be compared to SDEdit (Meng et al., 2021), which injects Gaussian noise for image editing and is commonly adopted in many image editing pipelines. Indeed, PnR demonstrates that similar yet general principles can be applied in parameter space for 3D editing.

While PnR currently focuses on static 3D scenes, future research could extend the proposed methods to 4D neural fields representing dynamic scenes. This extension would enable powerful video editing applications, such as modifying the motion of objects or characters in a 3D-consistent way while preserving their appearance and physical plausibility.

References

[1] Changyou Chen, Ruiyi Zhang, Wenlin Wang, Bai Li, and Liqun Chen. A unified particle-optimization framework for scalable bayesian sampling. *arXiv preprint arXiv:1805.11659*, 2018. 1

[2] Hanze Dong, Xi Wang, Yong Lin, and Tong Zhang. Particle-based variational inference with preconditioned functional gradient flow. *arXiv preprint arXiv:2211.13954*, 2022. 1

[3] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. *arXiv preprint arXiv:2303.12789*, 2023. 4, 6, 7

[4] Susung Hong, Donghoon Ahn, and Seungryong Kim. Debiasing scores and prompts of 2d diffusion for view-consistent text-to-3d generation. *Advances in Neural Information Processing Systems*, 36:11970–11987, 2023. 4

[5] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023. 5

[6] Juil Koo, Chanho Park, and Minhyuk Sung. Posterior distillation sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13352–13361, 2024. 4, 6, 7

[7] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6517–6526, 2024. 5

[8] Chang Liua and Jun Zhub. Geometry in sampling methods: A review on manifold mcmc and particle-based variational inference methods. *Advancements in Bayesian Methods and Implementations*, 47:239, 2022. 1

[9] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 6

[10] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 1, 6, 7

[11] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 6, 8

[12] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 5, 8

[13] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023. 6

[14] Ziyu Wang, Tongzheng Ren, Jun Zhu, and Bo Zhang. Function space particle optimization for bayesian neural networks. *arXiv preprint arXiv:1902.09754*, 2019. 1

[15] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36, 2024. 1